NASA-CR-203055

ISI Reprint Series
ISI/RS-92-421
Spring 1992

NDB

Prospero: A Tool for Organizing Internet Resources

7N-62-CR
0541

B. Clifford Neuman

ISI/RS-92-421
Spring 1992

University of Southern California
Information Science Institute
4676 Admiralty Way, Marina del Rey, CA 90292-6695
310-822-1511

This research was supported in part by the National Science Foundation (Grant No. CCR-8619663), the Washington Technology Centers, Digital Equipment Corporation, and the Defense Advanced Research Projects Agency under NASA Cooperative Agreement NCC-2-539. The views and conclusions contained in this article are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any of the funding agencies

Reprinted with permission from *Electronic Networking: Research, Applications and Policy,* Volume 2, Issue 1, Spring 1992..

Prospero: A Tool for Organizing Internet Resources

B. Clifford Neuman

Recent growth of the Internet has greatly increased the amount of information that is accessible and the number of resources that are available to users. To exploit this growth, it must be possible for users to find the information and resources they need. Existing techniques for organizing systems have evolved from those used on centralized systems, but these techniques are inadequate for organizing information on a global scale.

This article describes Prospero, a distributed file system based on the Virtual System Model. Prospero provides tools to help users organize Internet resources. These tools allow users to construct customized views of available resources, while taking advantage of the structure imposed by others. Prospero provides a framework that can tie together various indexing services producing the fabric on which resource discovery techniques can be applied.

The Internet contains a massive amount of information, but it is hard to use that information. There are several barriers to usability: it is difficult to identify the information of interest; it is difficult to keep track of this information once found; it is difficult to share information about what is available, or to collaboratively maintain such meta-information; and the information is often scattered across multiple file systems of different types, meaning that different mechanisms are needed to access it. Existing methods for organizing information have evolved from techniques used on centralized systems and are inadequate for organizing information on a global scale.

Users look for information in many ways. They consult libraries, journals, professional society publications, mailing lists, indexing services, and other users. While these sources of meta-information are useful, it is still necessary for users to identify the source that can answer their query. Prospero provides a framework within which such meta-information (which I will refer to as directories) can be made available to users, and it provides the tools to allow directories from multiple sources to be combined in useful ways.

Prospero lets users create customized views of a global file system. This customization plays an important role in organizing information since there are many communities of users, and they do not share the same interests. By supporting multiple views of the available information, one can improve the ease with which one finds information that is likely to be of interest, while keeping less useful information out of the way where the user is less likely to trip over it.

A prototype of Prospero is available and has been used to organize information on Internet sites world-wide. Prospero-based applications are used on more than 7,500 systems in 29 countries on six continents.

Organizing, Not Just Searching

There are four areas where work is needed to help users obtain the information they need: retrieval, indexing, search, and organization. A number of recent

B. Clifford Neuman is a computer scientist at the Information Sciences Institute of the University of Southern California. The work described in this article was begun while completing his Doctorate at the University of Washington. Neuman may be reached at USC/ISI, 4676 Admiralty Way, Marina del Rey, CA 90292-6695, USA. Telephone +1 (310) 822-1511, email bcn@isi.edu.

This research was supported in part by the National Science Foundation (Grant No. CCR-8619663), the Washington Technology Center, Digital Equipment Corporation, and the Defense Advance Research Projects Agency under NASA Cooperative Agreement NCC-2-539.

The views and conclusions contained in this article are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any of the funding agencies.

ELECTRONIC NETWORKING, Vol. 2, No. 1, Spring 1992, pp. 30–37. © 1992 Meckler

systems have addressed the first three areas, yet the fourth has been greatly ignored. Users require all four functions if they are to obtain the information they need. Without work on organization, the other functions become less useful as a system grows.

Some recently distributed file systems support a global name space. The Andrew File System (Howard et al., 1988) is an example. Such file systems provide for the retrieval of files worldwide, yet they do little to help the user find files of interest. Such file systems have directories near the root named after organizations, with the next level usually naming individual users. Files on particular topics are scattered across the leaves of the tree, where they are difficult to find.

Indexing can help users find information that is scattered across a distributed system. In attribute-based naming (Peterson, 1988), a name is resolved by querying a database of the attributes associated with local resources. Similarly, the Wide Area Information Server (WAIS) maintains a full-text index of a collection of documents, allowing users to search for documents by specifying words that appear in the full text (Kahle & Medlar, 1991). The Semantic File System (Gifford et al., 1991) provides another example of indexing by maintaining an index for all files on a collection of file servers. Distributed indexing (Danzig et al., 1991) provides an alternative approach to indexing widely distributed information. Indices are maintained by topic, and a topical index can request that future updates to other indexes be propagated if they match certain criteria. The indices in the systems described so far cover only a subset of the files that are available globally. It is still necessary for the user to find the correct server to query (selecting the index to be used).

Although it is possible to construct indices that cover large collections of files, it is necessary to trade detail and completeness for manageable size. For example, the Archie database (Emtage & Deutsch, 1992) indexes files from certain directories on major Internet FTP sites. The index, however, is based only on file names, not the file contents or other attributes. Completeness is also limited since only files available by anonymous FTP on selected sites are included. Another problem is that many queries return much more information than most users are prepared to deal with. In many cases, the large number of items found obscures the few that are really of interest.

When resources of interest to a user are distributed across multiple systems, and when the directory information needed to discover such resources is scattered across multiple indices, resource discovery techniques are needed to search for the desired information. Simplistic search strategies such as global broadcast or exhaustive depth-first search (as used by the Unix find command) are not suitable for large systems. Instead, search techniques should be based on browsing: looking at the information presently available and expanding the search in directions most likely to yield the desired results. Such browsing might include an interactive dialogue with the user (as is the case for directory browsers), it might be highly automated while accepting input from the user to narrow the search (as is done in Schwartz and Tsirigotis' (1991) resource discovery work), or once initiated it might run independently, returning the results to the user [knowbots (Kahn & Cerf, 1988) fall into this class].

Such search strategies are useful primarily when information is organized in such a way that programs and users can determine the appropriate direction in which to expand a search. One way to do this is to build a hierarchical directory service that can be used to find indexing services with information on various topics. Dalton (1991) discusses the possibility of using X.500 for this purpose. Such an approach works best when organizing a limited number of objects or when a single administrator can decide what is to appear in the upper levels of the name space.

The X.500 approach breaks down administratively, however, if used to organize fine-grained objects on a global scale. It is very difficult to gain agreement on what topics should appear near the top of the tree, and once topics are agreed on, there is disagreement on which resources should be included under each topic. This problem is apparent on Usenet, a worldwide distributed message service for disseminating messages on many topics. A significant share of the messages sent on Usenet discuss what messages are appropriate for particular newsgroups, whether new newsgroups should be created, and what they should be called. This clearly demonstrates the problem of reaching consensus on globally shared names.

Instead of supporting a single hierarchy for organizing information, it is possible to allow each user to organize information on his or her own. This customization is important for a number of reasons: it reduces the clutter that would otherwise be caused by resources in which the user has little interest; it allows users to define shorter names for frequently referenced resources; and it allows users to replace entire portions of the naming hierarchy with alternative views more appropriate for their particular needs. User-centered naming also eliminates the need for consensus when deciding what should appear in the upper levels of the naming hierarchy. Each user can make that decision based on his or her own opinions.

Organizational mechanisms must make available directory information from many sources, including existing indexing schemes[1] and directory information specified by users. It should be possible for directory information from different sources to be combined in useful ways.

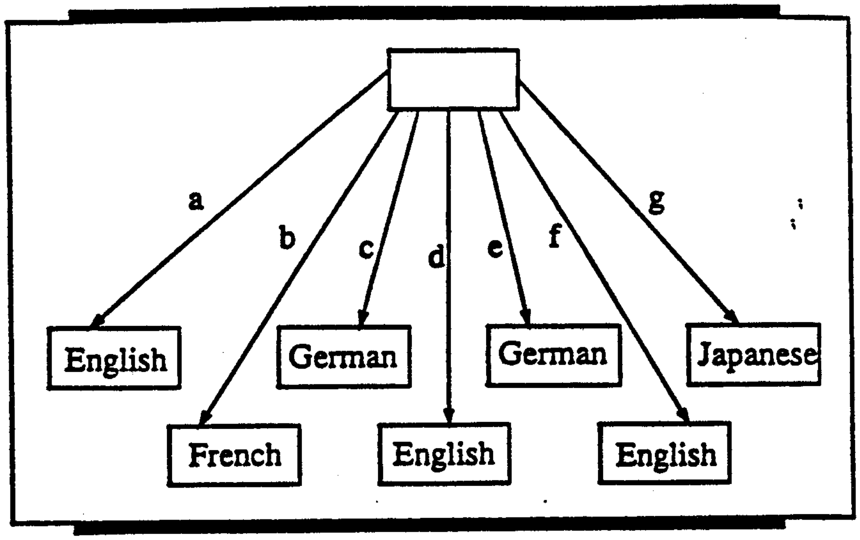

Figure 1. Directory before application of a filter

The Virtual System Model

The Virtual System Model (Neuman, 1992) provides a framework for organizing large systems within which users construct their own "virtual" systems by selecting objects and services that are available over the network; users then treat the selected resources as a single system, ignoring those resources that were not selected. The Prospero file system is a file system based on the Virtual System Model. By supporting a customized view of the system, information of interest to a user is prominently located near the center of the user's name space, while information that is not of interest is kept out of the way.

As users organize virtual systems for their own use, the structure imposed on the information can often be used by others. The Prospero naming network forms a generalized directed graph. A user's name space appears hierarchical and corresponds to the names seen by the user starting from a particular node in the graph, the root of the name space. If a user finds an object or a directory of interest, the user can add a link that will make the object more prominent. When a user creates a directory with links to objects on particular topics, others can (if authorized) view that directory and include it in their own virtual systems, thus benefiting from the organization imposed by the first user.

Indexing services are made available through Prospero by treating the results of a query as a virtual directory. Users can add links to the directories corresponding to particular indices, and even to directories that correspond to queries executed upon those indices.

Two features of Prospero allow new views of information to be derived from meta-information that already exists. If a union link is included in a directory, the contents of the directory that is the target of the link appear to be included in the directory containing the link. This allows a directory to incorporate directory information from other sources. When the original source changes, the changes will also be reflected in the directory incorporating that information.

When constructing views, users can also associate functions (filters) with links that allow the creation of derived views from views that already exist. For example, in Figure 1 files are named with the labels *a* though *g*. Associated with each file is an attribute list, one attribute of which is the language in which the text was written. The value of the language attribute is shown in the box representing the file. By attaching the distribute() filter to the directory link, a derived view is created within which the files appear to be distributed across subdirectories according to the value of the language attribute. The derived view is shown in Figure 2.

A filter can be an arbitrary program that takes a representation of a directory as an argument and returns the same. It can add links to a directory, delete links, change the names of links, and even define new filters that are to be applied when traversing links deeper in the hierarchy. As arbitrary programs, filters can access any information needed to perform their function. Typically, this information includes attributes of files and the contents of other directories, but it might involve reading files or performing database queries. Although users can write their own filters, it is expected that most will use the set already defined for them.

Organizing Information with Prospero

The Virtual System Model allows information to be organized in many ways, and many parties will play a role in doing so. Among the entities that will organize information will be individuals, professional societies, libraries, governments, commercial indexing services, or any collection of individuals sharing a common interest. An important feature of the model is that the same information can be organized in multiple ways.

The individual in the best position to organize the papers written by a particular author is that author. With Prospero, an author can maintain a directory referencing his or her own work, or at least that work which others should find. The incentive for doing so is visibility. The ease with which others can find one's writings affects the likelihood that those writings will be used. By maintaining one's own index of papers, one can also add cross-references to more recent work as it is completed.

The usefulness of such a directory is greatly enhanced when it is itself referenced from a higher level directory of authors. Such directories are maintained today in library card catalogs and in reader's guides to the literature, but the job of maintaining such directories is greatly simplified when implemented using Prospero; the maintainer of the higher level index would only have to update the directory when new authors are added. Once added, it is up to the authors themselves, or to individuals maintaining directories on behalf of the authors, to keep the list of the author's publications current.

Organizations like the ACM and the IEEE might each maintain a directory of topics in computer science and designate experts in each area to maintain the directory on that topic. Organizations in other fields, for example, the American Medical Association, might do the same. The custodians of particular topics could add references to worthwhile items as they are discovered. In cases where certain well-crafted queries on automatically maintained databases yield useful results, those queries can be encoded in filters, and the result added to the collection of topics as a virtual directory. Libraries could then maintain directories of general fields such as computer science, chemistry, and literature with links to the directories maintained by various organizations.

Users will build their own hierarchies of files by creating directories, subdirectories, and files of their own and by adding links to files, directories, and subdirectories created by others. Files that are frequently accessed by a user will probably have short names while names will be longer for objects of less interest. Because directories of other users will be accessible from the user's virtual system, the virtual system will probably contain files that a user has never accessed and might not even know about. These files, however, will be deep in the user's hierarchy.

If individuals do not like the way information is organized, they can organize it themselves, or they can find different experts whose views more closely match their own. They can completely customize their own name space so that their alternative view is used instead of the more accepted view. In fact, which view is the accepted view becomes more a matter of whose views more people adopt, rather than whose view is officially sanctioned.

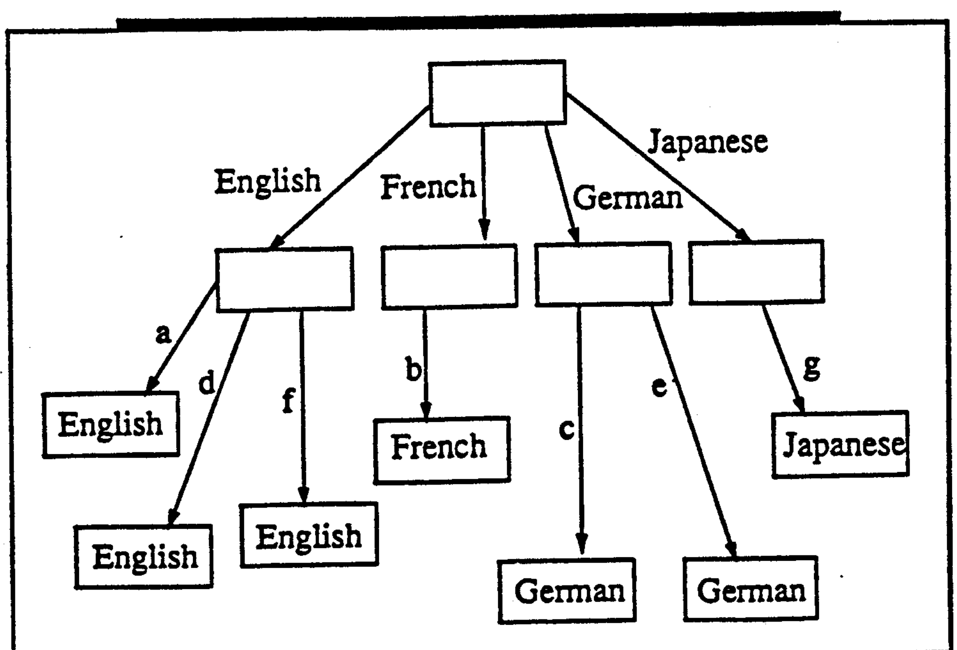

Figure 2. Directory with distribute() applied

Over time, multiple communities of users will evolve. It is expected that the members of each community will have similarly structured name spaces, but name spaces may vary widely across different communities of users. For example, members of the computer science community might organize virtual systems in one way, while members of the medical community might think of the world in a completely different manner.

Searching for Information

Once information has been organized, users can look for it in many ways. A user looking for a paper on heterogeneous computer systems by a particular author might find the paper in a directory maintained by that author. A user who did not know any of the authors might find the same paper in a directory of papers on distributed computing. Of course, just knowing that the information of interest exists in a published paper can be a big help; many times a user will not even know that.

Today, if something is available that is of interest, it is often found through directories such as the phone book or yellow pages, through reading newspapers and other periodicals, or by word of mouth. In the research community, these sources of information are supplemented by technical papers, electronic mail, and mailing lists. It is likely that these methods will continue to find significant use even once other mechanisms are in place. The Virtual System Model allows much of the information that is useful for finding objects, but which to date could only be obtained by external means (such as asking the author of a paper), to be included as part of the file system. The Prospero file system can then be used as the matrix through which users can navigate to find the desired information.

One way that information can be found using the Prospero directory service is through browsing. An individual interested in a particular topic can connect to the virtual system of someone else who is known to be interested in that topic.[2] The user could then look through those virtual systems for documents or files of interest. Of course, users would only see those files that the owner of the virtual system has authorized them to see.

Browsing is considerably more likely to be effective using Prospero than with traditional file systems. Prospero encourages users to make their own links to the files in which they have an interest. As such, interesting files are likely to appear in the hierarchies of many people, thus increasing the likelihood that the files will be found by browsing.

The Prospero directory service also provides the fabric on which resource discovery methods might operate. The Prospero server makes directory structures from existing systems part of that fabric, yet users can add their own links to augment the existing structure. Knowbots could navigate through the fabric and might themselves augment the existing structure by adding links to objects that they find. This augmentation of the naming network might provide both a method for a Knowbot to communicate its results back to its initiator, as well as a method through which knowbots can interact with each other.

Experience

A prototype of Prospero has been available since December 1990.[3] The prototype allows users to construct virtual systems and to navigate through them. In addition to the basic release, there are several standalone applications that rely on Prospero to retrieve directory information from indexing services.

Programs linked with the Prospero compatibility library are able to specify file names relative to the active virtual system when opening files. Prospero is a heterogeneous file system; instead of providing its own methods for accessing files, it relies on multiple underlying methods. The prototype presently supports Sun's Network File System, the Andrew File System, and the File Transfer Protocol (FTP). For FTP, the file is automatically retrieved, and the locally cached copy is then opened.

As distributed, a user's virtual system starts out with links to directories organizing information of various kinds in several ways. Figure 3 shows a sample session with Prospero. Users find information by moving from directory to directory in much the same manner as they would in a traditional file system. Users do not need to know where the information is physically stored. In fact, the files and directories shown in the example are scattered across the Internet. At any point, a user can access files in a virtual system as if they were stored on his or her local system.

In the example, the user connects to the root directory and lists it using the `ls` command. The result shows the categories of information included in the virtual system. The information includes online copies of papers (in the papers directory), archives of Internet and Usenet mailing lists (in the mailing-list and newsgroups directories), releases of software

```
Script started on Wed Jan 29 21:02:50 1992
% cd /
% ls
afs               info              papers
databases         lib               projects
documents         mailing-lists     releases
guest             newsgroups        sites
% cd papers
% ls
authors            conferences        subjects
bibliographies     journals           technical-reports
% cd technical-reports
% ls
Berkeley     IAState        OregonSt     UCalgary    UWashington
BostonU      MIT            Purdue       UColorado   Virginia
Chorus       NYU            Rochester    UFlorida    WashingtonU
Columbia     NatInstHealth  Toronto      UKentucky
Digital      OregonGrad     UCSantaCruz  UMichigan
% ls UCSantaCruz
crl
% ls UCSantaCruz/crl
ABSTRACTS.1988-89            ucsc-crl-91-01.ps.Z
ABSTRACTS.1990               ucsc-crl-91-02.part1.ps.Z
ABSTRACTS.1991               ucsc-crl-91-02.part2.ps.Z
ABSTRACTS.1992               ucsc-crl-91-02.ps.Z
INDEX                        ucsc-crl-91-03.ps.Z
ucsc-crl-88-28.ps.Z          ucsc-crl-91-06.ps.Z
...
% ls UWashington
cs   cse
%
% ls UWashington/cs
1991            INDEX          PRE-1991
1992            OVERALL-INDEX  README
% cd /papers
% ls
authors            conferences        subjects
bibliographies     journals           technical-reports
% ls journals
acm-sigcomm-ccr  ieee-tcos-nl
% ls journals/ieee-tcos-nl
app-form.ps.Z   v5n1           v5n3
cfp             v5n2           v5n4
% ls journals/acm-sigcomm-ccr
application.ps   jan89          jul90          sigcomm90-reg.ps
apr89            jan90          oct88
apr90            jan91          oct89
apr91            jul89          sigcomm90-prog.ps
% vls journals
   acm-sigcomm-ccr       NNSC.NSF.NET     /usr/ftp/CCR
   ieee-tcos-nl          FTP.CSE.UCSC.EDU /home/ftp/pub/tcos
%
script done on Wed Jan 29 21:06:53 1992
```

Figure 3. Sample session

packages (in the releases directory), and the contents of prominent Internet archive sites (in the sites directory). Files of interest can appear under more than one directory. For example, a paper that is available from a prominent archive site might also be listed under the papers directory.

Next, the user connects to the papers directory, lists it, and finds the available papers further categorized as conference papers, journal papers, or technical reports. The technical report directory is broken down by organization and by department within the organization. The journals directory is organized by the journal in which a paper appears, and the two journals that are shown are futher organized by issue. Use of the `vls` command shows where a file or directory is physically stored, demonstrating the fact that the files are scattered across the Internet (IEEE TC/OS Newsletter on FTP.CSE.UCSC.EDU and Computer Communications Review on NNSC.NSF.NET.) Though not shown in the example, papers are also organized by author and subject in other directories from the same virtual system.

It is important to note that the example shows only part of the information available through Prospero, and that it shows a typical way that the information is organized. Individuals can organize their own virtual systems differently.

One of the most frequently used directories in Prospero is that representing the Archie database, developed at McGill University (Emtage & Deutsch, 1992). That directory includes subdirectories organizing files according to the last components of their file names. For example, the subdirectory prosp contains references to the files available by Anonymous FTP whose names include the string *prosp*. Among the matches would be files related to Prospero. The contents of each subdirectory are equivalent to what would result from running the Unix `find` command with appropriate arguments over all the major archive sites on the Internet (if it were even possible to do so). The subdirectories do not exist individually but are instead created when referenced by querying the Archie database. The use of Archie through Prospero has been so successful that the Archie group has adopted Prospero as the preferred method for remote access to the Archie database.

To provide the benefits of Prospero to users who have not installed it on their systems, Steve Cliffe of the Australian Academic and Research Network (AARNet) Archive Working Group has added Prospero support to one of their FTP servers. As well as making files available from the physical file system, the modified FTP server makes files available from a virtual file system. When a retrieval request is received, the FTP server locates the file using Prospero and checks to see if a copy of the file is available locally. Using Prospero to check the last modified time of the authoritative copy, the FTP server checks that the local copy is current. If a current copy does not exist locally, the server retrieves and caches a copy of the file. The local copy is then returned to the client.

Future Plans

Prospero is an evolving system. We are continuing to work closely with the Archie group to make additional databases available. Immediate plans for the future also involve integrating Prospero with additional indexing services including WAIS (Kahle & Medlar, 1991), and once they are deployed, semantic file systems (Gifford et al., 1991) and distributed indices (Danzig et al., 1991). This will be accomplished by allowing a Prospero server to make meta-information from these databases available using the Prospero protocol.

In many respects, the goals of Prospero are similar to those of Hypertext systems such as World Wide Web (Berners-Lee et al., 1992). We hope to make information from that system available through Prospero.

We will be adding additional methods for retrieval of data. This will be of use when integrating WAIS and Prospero since much of the data indexed by WAIS is retrievable only with Z39.50. In addition to adding real-time access methods, we will be adding several off-line methods. For files that are accessible only by electronic mail, an e-mail method will be added that will automatically request the file on the user's behalf, allowing references to such files to be organized together with other files.

We also plan to add support for publications that are available only on paper. Indices for such information can be made available by running a Prospero server over a bibliographic database. The references would indicate the information needed to obtain a copy of the document, either an ISBN number or perhaps the shelf location in the local library.

Concluding Remarks

The Virtual System Model provides a powerful framework within which information can be organized. Prospero makes that framework available for organizing information on the Internet. By themselves, neither the model nor the prototype helps users find information of interest. Their contributions are in encouraging and enabling users to organize information in ways that make it easier to find things.

Professional societies, libraries, governments, commercial indexing services, and others will play important roles in organizing the information available from future systems. The Virtual System Model allows such service providers to build on each other's work, eliminating duplicated effort, and it allows users to construct views of the information provided by these services which better meet their own requirements. The real contribution of this work will depend on the extent to which the model is adopted by these service providers and how it is used in future systems.

Acknowledgments

Ed Lazowska provided valuable guidance throughout this work. Discussions with John Zahorjan, Hank Levy, and Alfred Spector helped me refine the ideas that ultimately led to the development of Prospero. Celeste Anderson, Ben Britt, Steve Cliffe, Peter Danzig, Peter Deutsch, Alan Emtage, Deborah Estrin, Dennis Hollingworth, and Charles McClure provided comments on earlier drafts of this paper.

Notes

1. In fact, indexing is itself a method for organizing information, although it is typically applied to only a subset of the information available.

2. The directories and files that a user maintains will be owned by that user. Parts of a user's hierarchy, however, may be owned by other users. Access control information is maintained along with each file or directory, and with each directory link. This information determines who is allowed to read the file or search the directory. It is expected that users will make parts of their hierarchies accessible to others, but how much is to be made available will be decided by the individual.

3. For information on obtaining the release please send a message to info-prospero@isi.edu.

References

Berners-Lee, Tim, Cailliau, Robert, Groff, Jean Francois, & Pollermann, Bernd. (1992). World-wide web: The information universe. *Electronic Networking: Research, Application, and Policy*, 2(1), 53–59.

Dalton, Marian L. (1991). Does anybody have a map? Accessing information in the Internet's virtual library. *Electronic Networking: Research, Application, and Policy*, 1(1), 31–39.

Danzig, Peter B., Ahn, Jongsuk, Noll, John, & Obraczka, Katia. (1991). Distributed Indexing: A scalable mechanism for distributed information retrieval. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 220–229). New York: ACM Press.

Emtage, Alan, & Deutsch, Peter. (1992). Archie: An electronic directory service for the Internet. *In Proceedings of the Winter 1992 USENIX Conference* (pp. 93–110). Berkeley, CA: USENIX Association [2560 Nonth St., Suite 215, Berkeley, CA 94710].

Gifford, David K., Jouvelot, Pierre, Sheldon, Mark A., & O'Toole, James W. Jr. (1991). Semantic File Systems. *Proceedings of the 13th ACM Symposium on Operating Systems Principles* (pp. 16–25). New York: ACM Press.

Howard, John H., Kazar, Michael L., Menees, Sherri G., Nichols, David A., Satyanarayanan, M., Sidebotham, Robert N., & West, Michael J. (1988). Scale and performance in a distributed file system. *ACM Transactions on Computer Systems*, 6(1), 51–81.

Kahle, Brewster, & Medlar, Art. (1991). *An information system for corporate users: Wide area information systems*. Technical Report TMC-199. Menlo Park, CA: Thinking Machines Corporation.

Kahn, Robert E., & Cerf, Vinton G. (1988). *The Digital Library Project; Volume 1: The world of knowbots* (Draft). Available from Corporation for National Research Initiatives, 1895 Preston White Dr., Suite 100, Reston, VA 22091.

Neuman, B. Clifford. (1992). *The Virtual System Model: A scalable approach to organizing large systems.* Doctoral dissertation (in preparation). Seattle,WA: University of Washington: Department of Computer Science and Engineering.

Peterson, Larry L. (1988). The Profile naming service. *ACM Transactions on Computer Systems*, 6(4), 341–364.

Schwartz, Michael F., & Tsirigotis, P. G. (1991). Experience with a semantically cognizant Internet White Pages Directory Tool. *Journal of Internetworking: Research and Experience*, 2(1), 25–50.